

Instituto Tecnológico de Costa Rica Escuela de Ingeniería en Computadores Programa de Licenciatura en Ingeniería en Computadores Curso: CE-4302 Arquitectura de Computadores II Profesores: Luis Alonso Barboza Artavia Ronald García Fernández Semestre: II 2025	Taller 03 CUDA Fecha de asignación: 04/11/25 (grupo 02) 05/11/25 (grupo 01) Fecha de entrega: 20/11/25 (grupo 02) 21/11/25 (grupo 01) Grupos de trabajo: 2 personas (máximo)
--	---

1. Descripción

En el procesamiento de texto, una operación fundamental es la conversión de caracteres entre minúsculas y mayúsculas.

En la codificación ASCII, los caracteres alfabéticos tienen valores numéricos específicos, La conversión de una letra minúscula a su equivalente en mayúscula se puede realizar restando un valor fijo. Este taller busca que las personas estudiantes mediante el uso de CUDA implementen un conversor sencillo de minúsculas a mayúsculas y viceversa.

2. Requisitos

- a- Google Colab o equivalente (capacidad de utilizar GPU)
- b- Opcional: GPU Nvidia

3. Investigue

A continuación, se formulan una serie de preguntas de guía para investigar sobre la implementación de algoritmos empleando intrinsics y extensiones SIMD.

- a- ¿Qué es CUDA?
- b- ¿Qué papel cumple un kernel en el modelo de programación CUDA y cómo se relaciona con los hilos de ejecución?
- c- ¿Qué significan los términos bloque, hilo y *grid* en CUDA, y cómo determinan la forma en que se reparte el trabajo en la GPU?
- d- ¿Qué son las funciones `cudaMalloc`, `cudaMemcpy`, `cudaFree`? ¿Qué parámetros reciben? ¿Por qué son importantes y qué consideraciones tienen?
- e- ¿Cómo se compila y ejecuta un programa CUDA?
- f- ¿Qué tipo de problemas en sistemas empujados se benefician más del paralelismo de CUDA y por qué?

4. Ejercicio de aplicación en CUDA

Para efectos de este taller se le brinda el siguiente ejercicio el cual tiene que ser resuelto de forma práctica utilizando CUDA en Google Colab o GPU (Nvidia):

En el procesamiento de texto, una operación fundamental es la conversión de caracteres entre minúsculas y mayúsculas.

En la codificación ASCII, los caracteres alfabéticos tienen valores numéricos específicos. La conversión de una letra minúscula a su equivalente en mayúscula se puede realizar restando un valor fijo de 0x20, como se muestra en la tabla 1.

Caracter	Valor Hexadecimal
'A'	0x41
'Z'	0x5A
'a'	0x61
'z'	0x7A
('a' - 'A')	0x20
('z' - 'Z')	0x20

Tabla 1. Ejemplo de relación de caracteres ASCII mayúscula y minúscula

De acuerdo con esta información, se solicita lo siguiente:

- Implemente un generador de cadenas de texto aleatorias (UTF-8) donde se puede especificar el tamaño, en donde se pueda configurar como están alineados los datos, **ambos** deben poder ser definidos en tiempo de ejecución, el generador debe ser capaz de definir un porcentaje de caracteres alfabéticos (0-100%).
- Basándose en lo realizado en el taller 2 del curso, implemente un algoritmo serial que resuelva el problema de conversión mayúscula – minúscula y viceversa, llame a este archivo **case_converter_serial.cpp**, incluya lo necesario para medir su desempeño (tiempo de ejecución, memoria, y cualquier otro aspecto que considere relevante y que sea posible capturar).
- Implemente el algoritmo mediante el uso de CUDA en un archivo **case_converter_SIMD.cu** **debe ser capaz de manejar datos no alineados**.
- Valide la correctitud de los resultados de la versión CUDA usando la misma cadena en ambos casos.
- Empleando el generador de cadenas de texto realice mediciones sobre ambas implementaciones, con diferentes tamaños de cadenas, porcentajes de caracteres alfabéticos y alineamientos (al menos 50 diferentes tamaños, 2 alineamientos y 10 diferentes porcentajes) tiene que justificar el rango de valores empleados en función de la plataforma y soporte de *CUDA*. (sugerencia implemente un script para automatizar esta tarea y realizar

gráficas ver figura 2).

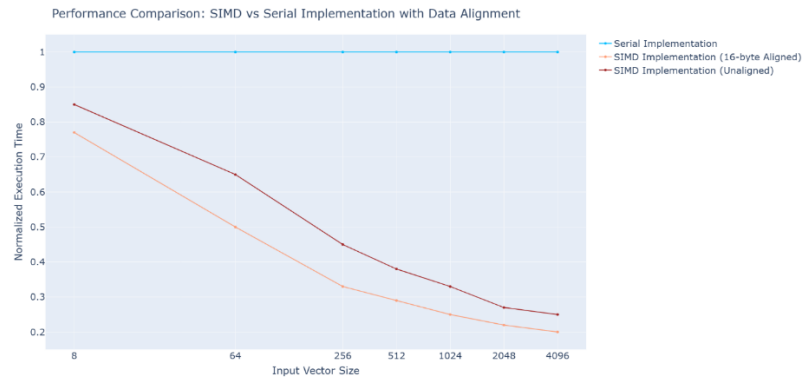


Figura 2. Ejemplo de resultados de tiempo de ejecución de algoritmos de conversión mayúscula y minúscula con un porcentaje de caracteres alfabéticos de 20%.

5. Justifique

Basándose en la implementación del punto 5 y lo investigado en este taller:

- ¿Cómo afecta el alineamiento de los datos el uso de CUDA?
- Realice un análisis comparativo de las gráficas de desempeño al aumentar el tamaño de la cadena de caracteres en ambas implementaciones y el porcentaje de caracteres alfabéticos (serial y CUDA)

6. Entregables

- Un documento **PDF** donde muestre las respuestas a las preguntas planteadas en la sección 3, junto con los resultados de la sección 4 y 5.
- En un archivo taller_CUDA.zip el código fuente (Notebook de Colab o archivos si utilizó GPU) con las implementaciones del punto 4, un archivo **README** con los detalles para ejecutar su código y graficar sus resultados. NO incluya ejecutables de ser así se le asignará un 0 de nota.

Si tienen dudas puede escribir al profesor al correo electrónico. Los documentos serán sometidos a control de plagios. La entrega se debe realizar por medio del TEC-Digital en la pestaña de evaluación.

No se aceptan entregas extemporáneas después de la fecha de entrega a las 23:59 como máximo.